DB Name	Query	Hit Count	Set Name
USPT	l1 and (variable with task$ with priority)	7	L6
USPT	l1 and (determin$ with variable with task$ with priority)	2	L5
USPT	l1 and ((determin$ with task$ with priority) same (fifo))	1	L4
USPT	l1 and ((task$ with priority) same (round with robin))	4	L3
USPT	l1 and (task$ with priority with round with robin)	2	L2
USPT	((712/$)!.CCLS.)	6983	L1

4,642,756

5,682,574

WEST

Freeform Search

Database:
US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: l1 and ((task$ with priority) same (round with robin))

Display: 10 **Documents in Display Format:** KWIC **Starting with Number** 1

Generate: ○ Hit List ◉ Hit Count ○ Image

Search | Clear | Help | Logout | Interrupt

Main Menu | Show S Numbers | Edit S Numbers | Preferences

Search History

Today's Date: 10/30/2001

DB Name	Query	Hit Count	Set Name
USPT	l1 and ((task$ with priority) same (round with robin))	4	L3
USPT	l1 and (task$ with priority with round with robin)	2	L2
USPT	((712/$)!.CCLS.)	6983	L1

L3

WEST

Generate Collection

L3: Entry 4 of 4 File: USPT Feb 10, 1987

DOCUMENT-IDENTIFIER: US 4642756 A
TITLE: Method and apparatus for scheduling the execution of multiple processing tasks in a computer system

BSPR:
Various task schedulers have been proposed and implemented which are somewhat adequate for servicing real-time or background type tasks. One approach in wide use is to assign fixed priorities to each task and base the scheduling of tasks solely on these priorities. While this approach may be suited for dedicated real-time applications, it is unacceptable for interactive applications where the nature of the interactive request is more important than a previously specified priority. Another approach is to use a fixed "time-slice" to direct the CPU to one task after another in a round-robin fashion. This approach has the effect of giving all tasks equal priority. Neither of these scheduling methods is adequate for a computer system which supports real-time, interactive, and background tasks.

CCXR:
712/244

WEST

Generate Collection

L7: Entry 2 of 2 File: USPT Feb 10, 1987

DOCUMENT-IDENTIFIER: US 4642756 A
TITLE: Method and apparatus for scheduling the execution of multiple processing tasks in a computer system

BSPR:
Two major benefits of the task scheduler of the invention are (1) the same computer system can simultaneously provide support to tasks which require fixed priorities relative to other tasks (such as real-time tasks) along with interactive tasks whose scheduling is dependent on the type of interaction; and (2) the method for assigning internally calculated priorities is able to select the correct task for execution in such a fashion as to optimize the responsiveness of the computer to a set of interactive computer users whose computational needs vary from moment to moment.

CCXR:
712/244

WEST

Generate Collection

L4: Entry 1 of 1 File: USPT Oct 28, 1997

DOCUMENT-IDENTIFIER: US 5682554 A
TITLE: Apparatus and method for handling data transfer between a general purpose computer and a cooperating processor

DEPR:
When down counter 25 reaches zero, it asserts its ZERO flag which deasserts the CYCLE.sub.-- STALL.sub.-- HC signal applied to host computer 22, and in step 319 the INTERRUPT.sub.-- HC signal is sent to host computer 22. The INTERRUPT.sub.-- HC signal then causes host computer 22 to execute the conventional interrupt handling routines, thereby interrupting its current data transfer. During the interrupt, host computer 22 executes its conventional interrupt routine that includes appropriate exception handling such as switching to a new process or polling FIFO 24 to determine if the ALMOST.sub.-- FULL signal is deasserted. Steps 321, 323 and 325 show the typical process for handling the interrupt from interface 23. In step 321, the system (typically the host computer's operating system) determines whether a process switch (i.e. switching of processing to a task other than data writing to FIFO 24) is appropriate; this is typically managed by the conventional operating system (e.g. UNIX) process handling routines, which assign a task/process identification (ID) to each task/process often with a priority for the task/process, and the tasks/processes operating within the system are "scheduled" for execution by the host computer 22. If the operating system determines that a process switch should occur, then other tasks/processes are executed in step 323 (again according the conventional operating system task scheduling procedures) and when the scheduling routines specify returning to the process of writing data to FIFO 24, then processing proceeds to step 325. If a process switch is not selected in step 321 then step 325 follows step 321. In step 325, the system polls FIFO 24 by checking the status of the ALMOST.sub.-- FULL signal. If the FIFO is no longer ALMOST.sub.-- FULL then processing returns (via node A) to step 301; otherwise (FIFO is still ALMOST.sub.-- FULL) processing loops back to step 321.

CCXR:
712/34

DB Name	Query	Hit Count	Set Name
USPT	l5 and (trace$ with interrupt$)	127	L6
USPT	((714/$)!.CCLS.)	18077	L5
USPT	l3 and (redundant$)	10	L4
USPT	l1 and (interrupt$ with latency)	103	L3
USPT	l1 and (interrupt$ same (redundant$ with state$ with register$))	0	L2
USPT	((712/$)!.CCLS.)	6983	L1

WEST

Generate Collection

L6: Entry 104 of 127 File: USPT Dec 19, 1989

DOCUMENT-IDENTIFIER: US 4888691 A
TITLE: Method for disk I/O transfer

BSPR:
The disk driver includes a trace function that can be selectively enabled for tracking system operations and errors. When the trace function is enabled, a real time entry is made in a trace table for each command block, each interrupt and each error that is handled by the disk driver. The trace table thus contains a traffic record useful for diagnostic purposes.

DEPR:
When the trace routine is first enabled, diagnostic checks of system performance are made. The trace table entries then are accumulated as part of each interrupt service routine, during the process user request routine and during the process clock routine. The trace routine operates only if the trace switch is enabled or on. Each time the disk controller initiates an interrupt, a trace table entry is made. The trace message in the entry indicates the interrupt type. Each time an interrupt service routine does not find a specified queue block in the pending queue, a trace table entry is made. The trace message in the entry indicates the type of interrupt and the fact that the specified ID was not found in the pending queue. In the case of an unsolicited interrupt by the disk controller, the trace queue pointer and trace ID are both zero since the unsolicited interrupt is not associated with a user request. The trace message in the table entry indicates the type of unsolicited interrupt. When a command block is built, a trace table entry is made. The trace message in the entry indicates the type of command. When the process clock routine is being executed, a trace table entry is made if a controller time out is detected. The trace message in the entry indicates that the controller has timed out. When a trace function is specified in the disk driver routines shown in FIGS. 9A-18 and described hereinafter, a trace table entry is made, but only if the trace switch is enabled.

DEPR:
A flow diagram of the UNSOLICITED interrupt service routine is shown in FIG. 16. The UNSOLICITED interrupt type is identified in step 802. When a DISK SPIN UP or a DISK SPUN DOWN interrupt is received, the information is posted to the mailbox in step 804. When a CONTROLLER ERROR or CONTROLLER REQUIRES WARM START interrupt is received, the controller is aborted and a trace table entry is made in step 806. When a SELF CHECKPOINT REQUIRED interrupt is received, the checkpoint flag is set in step 808. When the SELF CHECKPOINT COMPLETE interrupt is received, the checkpoint flag is reset in step 810.

CLPV:
making an entry in the trace table for each interrupt by the disk memory unit, and

CCOR:
714/15

WEST

Generate Collection

L6: Entry 104 of 127 File: USPT Dec 19, 1989

DOCUMENT-IDENTIFIER: US 4888691 A
TITLE: Method for disk I/O transfer

BSPR:
The disk driver includes a trace function that can be selectively enabled for tracking system operations and errors. When the trace function is enabled, a real time entry is made in a trace table for each command block, each interrupt and each error that is handled by the disk driver. The trace table thus contains a traffic record useful for diagnostic purposes.

DEPR:
When the trace routine is first enabled, diagnostic checks of system performance are made. The trace table entries then are accumulated as part of each interrupt service routine, during the process user request routine and during the process clock routine. The trace routine operates only if the trace switch is enabled or on. Each time the disk controller initiates an interrupt, a trace table entry is made. The trace message in the entry indicates the interrupt type. Each time an interrupt service routine does not find a specified queue block in the pending queue, a trace table entry is made. The trace message in the entry indicates the type of interrupt and the fact that the specified ID was not found in the pending queue. In the case of an unsolicited interrupt by the disk controller, the trace queue pointer and trace ID are both zero since the unsolicited interrupt is not associated with a user request. The trace message in the table entry indicates the type of unsolicited interrupt. When a command block is built, a trace table entry is made. The trace message in the entry indicates the type of command. When the process clock routine is being executed, a trace table entry is made if a controller time out is detected. The trace message in the entry indicates that the controller has timed out. When a trace function is specified in the disk driver routines shown in FIGS. 9A-18 and described hereinafter, a trace table entry is made, but only if the trace switch is enabled.

DEPR:
A flow diagram of the UNSOLICITED interrupt service routine is shown in FIG. 16. The UNSOLICITED interrupt type is identified in step 802. When a DISK SPIN UP or a DISK SPUN DOWN interrupt is received, the information is posted to the mailbox in step 804. When a CONTROLLER ERROR or CONTROLLER REQUIRES WARM START interrupt is received, the controller is aborted and a trace table entry is made in step 806. When a SELF CHECKPOINT REQUIRED interrupt is received, the checkpoint flag is set in step 808. When the SELF CHECKPOINT COMPLETE interrupt is received, the checkpoint flag is reset in step 810.

CLPV:
making an entry in the trace table for each interrupt by the disk memory unit, and

CCOR:
714/15